

LDPC Lattice Codes for Full-Duplex Relay Channels

Hassan Khodaiemehr, Dariush Kiani and Mohammad-Reza Sadeghi

Abstract

Low density parity check (LDPC) lattices are obtained from Construction D' and a family of nested binary LDPC codes. We consider an special case of these lattices with one binary LDPC code as underlying code. This special case of LDPC lattices can be obtained by lifting binary LDPC codes using Construction A lattices. The LDPC lattices were the first family of lattices which have efficient decoding in high dimensions. We employ the encoding and decoding of the LDPC lattices in a cooperative transmission framework. We establish two efficient shaping methods based on hypercube shaping and Voronoi shaping, to obtain LDPC lattice codes. Then, we propose the implementation of block Markov encoding for one-way and two-way relay networks using LDPC lattice codes. This entails owning an efficient method for decomposing full-rate codebook into lower rate codebooks. We apply different decomposition schemes for one-way and two-way relay channels which are the altered versions of the decomposition methods of low density lattice codes (LDLCs). Due to the lower complexity of the decoding for LDPC lattices comparing to LDLCs, the complexity of our schemes are significantly lower than the ones proposed for LDLCs. The efficiency of the proposed schemes are presented using simulation results.

Keywords

Relay networks, lattice codes, block Markov encoding.

I. INTRODUCTION

Hassan Khodaiemehr, Dariush Kiani and Mohammad-Reza Sadeghi are with the Department of Mathematics and Computer Science, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. (Emails: {h.khodaiemehr, dkiani, msadeghi}@aut.ac.ir).

Part of this work was presented in [1] at IWCIT 2015, Iran.

OVER the last few years, cooperative transmission has become widely prominent with the increases in the size of communication networks. In wireless networks, the transmitted message from a node is heard not only by its intended receiver, but also by other neighbour nodes and those neighbour nodes can use the received signals to help transmission. They bring a cooperative transmission by acting as relays.

The relay channel first introduced by van der Meulen in [2] and it consists of a source aiming to communicate with a destination with the help of a relay. In this case, we call the relay channel *one-way relay channel* or *single relay channel*. In [3], Cover and El Gamal proposed the fundamental *decode-forward* (DF) and *compress-forward* (CF) schemes for the one-way relay channels which achieve near capacity rates. In DF scheme, the relay decodes the messages from the source and forwards them to the destination. In CF scheme, the relay compresses received signals and forwards the compression indices.

It is proved that the DF scheme is optimal for these types of channels: for physically degraded relay channels [3] in which the output observed at the receiver is a degraded version of the channel output at the relay, for semi-deterministic channels [4] in which the channel output at the relay is a deterministic function of the channel input at the transmitter and the channel input at the relay. The exact capacity of general relay channels is not known to date, although, there exist tight capacity approximations for a large class of networks [5], [6], and schemes like DF and CF achieve near-capacity rates. The upper bound on capacity is given by the cut-set upper bound [3] and the lower bound is given by Chong et al. in [7]. The scheme in [7] is a block-Markov transmission scheme that is a combination of the DF scheme and the CF scheme.

The one-way relay channel can be extended to the *two-way relay channel*, where a relay helps two users exchange their messages. Two types of two-way relay channels can be considered, that is, without a direct link between the two users, and with a direct link between the two users. The former is a suitable model for wired communication and the latter is suitable for wireless communication. Applications of relay cooperation can be seen in increasing the capacity [8], combating the fading effect [9], mitigating the effects of interference [10], [11], [12] and increasing the physical layer security [13]. However, DF scheme has been used in numerous applications, it achieves capacity only in special few cases. All of these approaches are using random Gaussian coding which is impractical for implementation. Thus, applying DF scheme in a practical scenario is interesting. One of the research areas that has such potential is lattice

theory.

An n dimensional lattice in \mathbb{R}^m is the set of integer linear combinations of a given basis with n linearly independent vectors in \mathbb{R}^m . Using lattices for communication over the real AWGN channel, has been investigated by Poltyrev [14]. In such a communication system, instead of the coding rate and capacity, normalized logarithmic density (NLD) and generalized capacity C_∞ have been introduced, respectively. Using Construction D of lattices [15], the existence of sphere-bound-achieving and capacity-achieving lattices has been proved by Forney et al. [16]. A capacity-achieving lattice code can be obtained from a capacity-achieving lattice together with a proper shaping region [17], [18]. Lattice codes are the Euclidean-space analog of linear codes. Researchers have also studied practical lattice codes. The search for practical implementable capacity-achieving lattices and lattice codes started by proposing low density parity check (LDPC) lattices [19]. In this class of lattices, a set of nested LDPC codes and Construction D' of lattices [15] are used to generate lattices with sparse parity check matrices. Another class of lattices, called low density lattice codes (LDLC), introduced and investigated in [20]. Turbo lattices employed Construction D along with turbo codes to achieve capacity gains [21]. Low density Construction A (LDA) lattices [22] are another class of lattices with near-capacity error performance and low-complexity, low-storage decoder. An LDA lattice can be obtained from Construction A [15], [23] with a non-binary LDPC code as its underlying code.

The use of lattice codes in relay networks has received significant attentions in recent years [24], [25], [26] [27], [28], [29]. It was shown in [27] and [28] that lattice codes can achieve the DF rates for the relay channel. All of these achievable schemes rely on asymptotic code lengths, which is a drawback in practical implementation. Recently, Aazhang et al. proposed a practical scheme based on LDLCs, for the real-valued, full-duplex one-way and two-way relay channels [28]. In this work, we propose another class of practical, efficient lattice codes, based on LDPC lattices, for the real-valued, full-duplex one-way and two-way relay channels.

The rest of this paper is organized as follows. Section II introduces the system models of the one-way and two-way relay channels. Section III presents the preliminaries on lattices and lattice codes. In Section IV, we introduce LDPC lattices. The encoding and decoding of these lattices are also presented in this section. In Section V, we consider the application of the LDPC lattices in the power constrained AWGN channels by presenting two efficient shaping methods, based on hypercube shaping and nested lattice shaping. In Section VI, we adapt our

shaping algorithms that enable us to do the decomposition of the LDPC lattices into lower-rate components without loss of shaping efficiency. Then, we present a practical block Markov scheme for the real-valued, full-duplex one-way relay channels, based on LDPC lattices. In Section VII, we present another decomposition method based on doubly nested LDPC lattices, for the two-way relay channels. Finally, in Section VIII, we examine the practical performance of our proposed schemes. Section IX contains the concluding remarks.

II. SYSTEM MODEL

A. One-Way Relay Channel

The relay channel that we have considered, is a three-terminal relay channel. First, we present the one-way relay channel, depicted in Fig.1 (a). The source transmits a message, which is mapped to a codeword, to both relay and destination and in the next time slot relay aids the destination by sending the part of the information of the previous time slot. We assume a full-duplex relay which can simultaneously transmit and receive the messages. For simplicity we suppose real-valued channels. Let \mathbf{x}_S and \mathbf{x}_R denote the signals transmitted by the source and

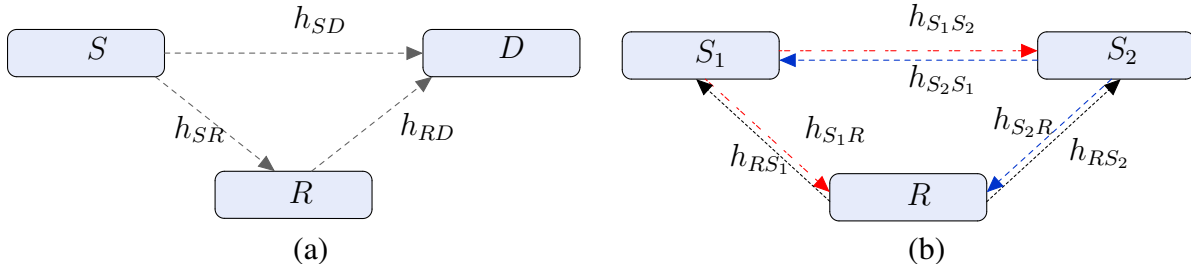


Fig. 1. (a) Full-duplex Gaussian one-way relay channel. (b) Full-duplex Gaussian two-way relay channel.

the relay. Let \mathbf{y}_R and \mathbf{y}_D denote the signals received at the relay and the destination. The received signals are

$$\mathbf{y}_R = h_{SR}\mathbf{x}_S + \mathbf{z}_R \quad (1)$$

$$\mathbf{y}_D = h_{SD}\mathbf{x}_S + h_{RD}\mathbf{x}_R + \mathbf{z}_D, \quad (2)$$

where $\mathbf{z}_R \sim \mathcal{N}(0, N_R)$, $\mathbf{z}_D \sim \mathcal{N}(0, N_D)$. Moreover, $h_{SR} = d_{SR}^{-\alpha_1}$, $h_{SD} = 1$, and $h_{RD} = d_{RD}^{-\alpha_2}$ are the channel gains between source, relay, and destination, which follow the usual path-loss

model. In particular, we take the distance between source and destination to be unity, d_{SR} and d_{RD} to be the distance between source to relay and relay to destination, and α_1 and α_2 to be the corresponding path-loss exponents. We constrain the source and relay transmissions to have average power no greater than P_S and P_R .

In general, the capacity of this channel is unknown however, the DF scheme which is proposed in [3], achieves the following inner bound:

$$R \leq \frac{1}{2} \min \left\{ \log_2 \left(1 + \frac{h_{SR}^2 P_S E\{\mathbf{x}_S^2\}}{N_R} \right), \log_2 \left(1 + \frac{h_{SD}^2 P_S E\{\mathbf{x}_S^2\} + h_{RD}^2 P_R E\{\mathbf{x}_R^2\}}{N_D} \right) \right\}.$$

This rate can be achieved via *block Markov encoding*. After decoding the message, the relay re-encodes the message and transmits the corresponding codeword in the next block. The lattice-coding version of block Markov encoding is proposed in [25]. It is proved theoretically that it can achieve the decode-and-forward rates. The results of [25] suggest that structured lattice codes may be used to outperform, random Gaussian codes in general Gaussian networks. The authors of [28] have applied this scheme and designed a family of practically implementable LDLC lattice codes for relay channels. In this paper we present another family of lattice codes which are amenable to practical implementation in block Markov schemes.

B. Two-Way Relay Channel

Next, we present the full-duplex Gaussian two-way relay channel, as depicted in Fig.1 (b). The two sources S_1 and S_2 exchange their messages, which are mapped to codewords and transmitted over the wireless medium. The relay node R overhears the noisy superposition of signals transmitted from sources and makes its own transmissions to facilitate communications. This channel can be modeled as follows

$$\mathbf{y}_R = h_{S_1R}\mathbf{x}_{S_1} + h_{S_2R}\mathbf{x}_{S_2} + \mathbf{z}_R, \quad (3)$$

$$\mathbf{y}_{S_1} = h_{S_2S_1}\mathbf{x}_{S_2} + h_{RS_1}\mathbf{x}_R, \quad (4)$$

$$\mathbf{y}_{S_2} = h_{S_1S_2}\mathbf{x}_{S_1} + h_{RS_2}\mathbf{x}_R, \quad (5)$$

where the noise components are $\mathbf{z}_R \sim \mathcal{N}(0, N_R)$, $\mathbf{z}_{S_1} \sim \mathcal{N}(0, N_{S_1})$ and $\mathbf{z}_{S_2} \sim \mathcal{N}(0, N_{S_2})$. Similar to the one-way relay channel model, $h_{S_1S_2}$, $h_{S_2S_1}$, h_{S_2R} , h_{S_1R} , h_{RS_1} and h_{RS_2} are the channel gains which follow the usual path-loss model. We constrain the sources S_1 , S_2 and relay R transmissions to have average powers no greater than P_{S_1} , P_{S_2} and P_R , respectively.

The capacity of this channel is unknown, but a DF scheme was presented in [25] that achieves rate pairs (R_1, R_2) satisfying (6),

$$R_i \leq \min \left\{ \frac{1}{2} \log_2 \left(1 + \frac{h_{RS_j}^2 P_R E \{ \mathbf{x}_R^2 \} + h_{S_i S_j}^2 P_{S_i} E \{ \mathbf{x}_{S_i}^2 \}}{N_{S_j}} \right), \right. \\ \left. \left[\frac{1}{2} \log_2 \left(\frac{h_{S_i R}^2 P_{S_i} E \{ \mathbf{x}_{S_i}^2 \}}{h_{S_i R}^2 P_{S_i} E \{ \mathbf{x}_{S_i}^2 \} + h_{S_j R}^2 P_{S_j} E \{ \mathbf{x}_{S_j}^2 \}} + \frac{h_{S_i R}^2 P_{S_i} E \{ \mathbf{x}_{S_i}^2 \}}{N_R} \right) \right]^+ \right\}, \quad (6)$$

in which $[x]^+ \triangleq \max \{x, 0\}$, $i, j \in \{1, 2\}$, and $i \neq j$.

III. PRELIMINARIES ON LATTICES

A discrete, additive subgroup Λ of the m -dimensional real space \mathbb{R}^m is a lattice. Every lattice Λ has a basis $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\} \subset \mathbb{R}^m$ where every $\mathbf{x} \in \Lambda$ can be represented as an integer linear combination of vectors in \mathcal{B} . The rank of the lattice is n and its dimension is m . If $n = m$, the lattice is called a full-rank lattice. In this paper, we consider full-rank lattices. The matrix \mathbf{M} with $\mathbf{b}_1, \dots, \mathbf{b}_n$ as rows, is a generator matrix for the lattice. The matrix $\mathbf{G} = \mathbf{M}\mathbf{M}^t$ is a Gram matrix for the lattice. The determinant of the lattice $\det(\Lambda)$ is defined to be the determinant of the matrix \mathbf{G} and the volume of the lattice is defined as $\text{vol}(\Lambda) = \sqrt{\det(\mathbf{G})}$. A Voronoi cell $\mathcal{V}(\mathbf{x})$ is the set of those points of \mathbb{R}^n that are at least as close to \mathbf{x} as to any other point in Λ . We call the Voronoi region associated with the origin, the fundamental Voronoi region of Λ , denoted by \mathcal{V} or $\mathcal{V}(\Lambda)$.

We say that a lattice Λ_s is nested in Λ_c if $\Lambda_s \subset \Lambda_c$. Using nested lattices in \mathbb{R}^n , define the codebook $\mathcal{C} = \mathcal{V}_s \cap \Lambda_c$ which has the rate

$$R = \frac{1}{n} \log_2(|\mathcal{C}|) = \frac{1}{n} \log_2 \left(\frac{\text{vol}(\mathcal{V}_s)}{\text{vol}(\mathcal{V}_c)} \right). \quad (7)$$

Suppose that the points of a lattice Λ are sent over an unconstrained AWGN channel with noise variance σ^2 . The volume-to-noise ratio (VNR) of lattice Λ is defined as

$$\text{VNR} = \frac{\text{vol}(\Lambda)^{\frac{2}{n}}}{2\pi e \sigma^2}. \quad (8)$$

For a large n , VNR is the ratio of the normalized volume of Λ to the normalized volume of a noise sphere of squared radius $n\sigma^2$ which is defined as generalized signal-to-noise ratio (SNR) in [19] and α^2 in [16].

Let $\mathbf{x} \in \Lambda$ be the transmitted vector on the unconstrained AWGN channel, then the received vector \mathbf{r} can be written as $\mathbf{r} = \mathbf{x} + \mathbf{e}$, where $\mathbf{e} = (e_1, \dots, e_n)$ and its components are independently and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ^2 . The probability of correct decoding, under maximum likelihood decoding, is given by

$$P_c = \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}(\mathbf{x})} e^{-\frac{\|\mathbf{t}\|^2}{2\sigma^2}} d\mathbf{t}, \quad (9)$$

where $\|\mathbf{x}\|$ is the Euclidean norm of \mathbf{x} .

A lattice quantizer is a map $\mathcal{Q}_\Lambda : \mathbb{R}^n \rightarrow \Lambda$ for some lattice $\Lambda \subset \mathbb{R}^n$. If we use the nearest-neighbor quantizer $\mathcal{Q}_\Lambda^{(NN)}$, then the quantization error $\mathbf{x}_e \triangleq \mathbf{x} - \mathcal{Q}_\Lambda^{(NN)}(\mathbf{x}) \in \mathcal{V}(\Lambda)$. Let \mathbf{x}_e be uniformly distributed over the Voronoi region $\mathcal{V}(\Lambda)$. Then, the *second moment per dimension* of Λ is

$$\sigma^2(\Lambda) = E[\|\mathbf{x}_e\|^2] = \frac{1}{n} \frac{1}{\text{vol}(\Lambda)} \int_{\mathcal{V}(\Lambda)} \|\mathbf{x}_e\|^2 d\mathbf{x}_e. \quad (10)$$

A *lattice constellation* $C(\Lambda, \mathcal{R}) = (\Lambda + \mathbf{t}) \cap \mathcal{R}$ is the finite set of points in a lattice translate $\Lambda + \mathbf{t}$ that lies within a compact bounding region \mathcal{R} of n -dimensional real space \mathbb{R}^n . The key geometric properties of the region \mathcal{R} are its *volume* $\text{vol}(\mathcal{R})$ and the *average energy* $P(\mathcal{R})$ per dimension of a uniform probability density function over \mathcal{R} (see, e.g. [30] and [15]):

$$P(\mathcal{R}) = \int_{\mathcal{R}} \frac{(\|\mathbf{x}\|^2/n) d\mathbf{x}}{\text{vol}(\mathcal{R})}. \quad (11)$$

The *normalized second moment* of \mathcal{R} is defined as

$$G(\mathcal{R}) = \frac{P(\mathcal{R})}{\text{vol}(\mathcal{R})^{2/n}}. \quad (12)$$

The normalized second moment of any n -cube centered at the origin is $1/12$. The *shaping gain* $\gamma_s(\mathcal{R})$ of \mathcal{R} , measures the decrease in average energy of \mathcal{R} relative to a baseline region, namely, an interval $[-d_0/2, d_0/2]$ or an n -cube $[-d_0/2, d_0/2]^n$, where d_0 is related to the $\text{vol}(\mathcal{R})$ [15]. The definition of shaping gain is

$$\gamma_s(\mathcal{R}) = \frac{\text{vol}(\mathcal{R})^{2/n}}{12P(\mathcal{R})} = \frac{1}{12G(\mathcal{R})}. \quad (13)$$

The optimum n -dimensional shaping region is an n -sphere [30]. The key geometrical parameters

of an n -sphere ($= \otimes$) of radius r for an even n are [15]:

$$\text{vol}(\otimes) = \frac{(\pi r^2)^{n/2}}{(n/2)!}, \quad (14)$$

$$P(\otimes) = \frac{r^2}{n+2}, \quad (15)$$

$$G(\otimes) = \frac{P(\otimes)}{\text{vol}(\otimes)^{2/n}} = \frac{((n/2)!)^{2/n}}{\pi(n+2)}. \quad (16)$$

The shaping gain of an n -sphere is a function of the dimension n . When n approaches infinity the shaping gain approaches the *ultimate shaping gain* $\pi e/6$ (1.53dB). The *shaping loss* $\lambda_s(\mathcal{R})$ of a shaping region \mathcal{R} with respect to an n -dimensional sphere, where n is even, based on (14)-(16), is [31]

$$\lambda_s(\mathcal{R}) = \frac{G(\mathcal{R})}{G(\otimes)} = \frac{\pi(n+2)G(\mathcal{R})}{\Gamma(\frac{n}{2}+1)^{2/n}}. \quad (17)$$

The shaping loss is greater than or equal to 1. If we form the intersection $\Lambda \cap \mathcal{R}$ of a lattice with a shaping region $\mathcal{R} \subset \mathbb{R}^n$, we would expect to obtain a code with about $\text{vol}(\mathcal{R})/\text{vol}(\Lambda)$ codewords [32]. In fact, by using Minkowski-Hlawka Theorem, it is proved that the value $\text{vol}(\mathcal{R})/\text{vol}(\Lambda)$ is correct in the average over a suitable set of lattices based on codes [32]. The rate of the code $\mathcal{C} = \Lambda \cap \mathcal{R}$ is approximately

$$R \approx \frac{1}{n} \log_2 \left(\frac{\text{vol}(\mathcal{R})}{\text{vol}(\Lambda)} \right). \quad (18)$$

IV. LDPC LATTICES

There exist many ways to construct lattices based on codes [15]. Here we mention two of them. The first one is Construction A and the other one is Construction D'. Assume that \mathcal{C} is a linear code over \mathbb{F}_p where p is a prime number, i.e. $\mathcal{C} \subseteq \mathbb{F}_p^n$. A lattice Λ based on Construction A [15] can be derived from \mathcal{C} as follows

$$\Lambda = p\mathbb{Z}^n + \epsilon(\mathcal{C}), \quad (19)$$

where $\epsilon: \mathbb{F}_p^n \rightarrow \mathbb{R}^n$ is the embedding function. In this work, we are particularly interested in lattices with $p = 2$.

Construction D' converts a set of parity checks defined by a family of nested codes $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \dots \supseteq \mathcal{C}_a$, into congruences for a lattice [15]. The number $a+1$ is called the level of the construction. An LDPC lattice $\Lambda \subset \mathbb{Z}^n$ can be constructed from Construction D' and a number

of nested binary LDPC codes. More detail about the structure and decoding of these lattices can be found in [19]. If we consider one code as underlying code of Construction D', which means $a = 0$, Construction A is obtained [23, Proposition 1]. In this case, Construction A LDPC lattices or 1-level LDPC lattices [23] will be obtained. In this paper, we refer to them as LDPC lattices without mentioning the level of the construction.

Definition 1: An LDPC lattice Λ is a lattice constructed based on Construction A or D' along with one binary LDPC code \mathcal{C} as its underlying code. Equivalently, $\mathbf{x} \in \mathbb{Z}^n$ is in Λ if and only if $\mathbf{H}\mathbf{x}^t = \mathbf{0} \pmod{2}$, where \mathbf{H} is the parity check matrix of \mathcal{C} .

The Generator matrix of Construction A lattice Λ using the underlying code $\mathcal{C} \subset \mathbb{F}_2^n$ is of the form

$$\mathbf{G}_\Lambda = \begin{bmatrix} \mathbf{I}_k & \mathbf{P}_{k \times (n-k)} \\ \mathbf{0}_{(n-k) \times k} & 2\mathbf{I}_{n-k} \end{bmatrix}, \quad (20)$$

where $\mathbf{G}_\mathcal{C} = \begin{bmatrix} \mathbf{I}_k & \mathbf{P} \end{bmatrix}$ is the generator matrix of \mathcal{C} in systematic form, k is the rank of \mathcal{C} , \mathbf{I}_k and $\mathbf{0}_k$, are identity and all zero square matrices of size k , respectively.

A. Encoding and Decoding of LDPC Lattices

The practical encoding and decoding of LDPC lattices, both with linear complexity in the dimension of the lattice, has been addressed in [33]. In this paper, we consider a translated and scaled version of the lattice Λ , generated by (20), as suggested in [34], [33] and [15, §20.5]. In the sequel, we present the decoding of these scaled and translated versions of LDPC lattices, which is proposed in [33] and it is obtained by combining the suggested decoding method in [15, §20.5], and the decoding of binary LDPC codes. Construction and decoding of these new lattices can be done using the following steps. First, convert the codewords of $[n, k]$ binary code \mathcal{C} into ± 1 notation (convert 0 to -1 and 1 to 1) [15, §20.5] which produces a set $\Lambda(\mathcal{C})$ consisting of the vectors of the form

$$\mathbf{c} + 4\mathbf{z}, \quad \mathbf{c} \in \mathcal{C}, \quad \mathbf{z} \in \mathbb{Z}^n. \quad (21)$$

The set of points in (21) strictly speaking, is not a lattice, but the translate of a lattice by the vector $(-1, -1, \dots, -1)$. The regular addition of vectors of the form (21) will not be of the same form. However, we can show that $\Lambda(\mathcal{C})$ is closed under following addition. For any

$\lambda_1, \lambda_2 \in \Lambda(\mathcal{C})$, we have [1], [33]

$$\lambda_1 \oplus \lambda_2 \triangleq \lambda_1 + \lambda_2 + (1, \dots, 1) \in \Lambda(\mathcal{C}). \quad (22)$$

The encoding of an integer row vector $\mathbf{b} \in \mathbb{Z}^n$ can be done as follows

$$\mathcal{E}(\mathbf{b}) = 2\mathbf{b}\mathbf{G}_\Lambda - (1, \dots, 1), \quad (23)$$

where \mathcal{E} is encoding function and \mathbf{G}_Λ is defined as (20). Let $\mathbf{x} = \mathbf{c} + 4\mathbf{z}$ be the transmitted lattice vector and \mathbf{y} be the received vector from AWGN channel

$$\mathbf{y} = \mathbf{c} + 4\mathbf{z} + \mathbf{n}, \quad (24)$$

where $\mathbf{c} \in \mathcal{C}$ and \mathcal{C} is a binary LDPC code in ± 1 notation, $\mathbf{z} \in \mathbb{Z}^n$ and $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$. First, we decode \mathbf{c} and next we find \mathbf{z} . The proposed algorithm in [33] is similar to the sum-product algorithm (SPA) for LDPC codes in message passing structure [35]. The inputs are the log likelihood ratios (LLR) for the a priori message probabilities from each channel. The estimation of the LLR vector $\gamma = (\gamma_1, \dots, \gamma_n)$ for LDPC lattices is proposed in [33] as follows

$$\gamma_i = \log \left(\frac{\Pr \{c_i = -1 | y_i\}}{\Pr \{c_i = +1 | y_i\}} \right) \triangleq 2 \left(\frac{(\frac{y_i-1}{4} - \lfloor \frac{y_i-1}{4} \rfloor)^2 - (\frac{y_i+1}{4} - \lfloor \frac{y_i+1}{4} \rfloor)^2}{\sigma^2} \right), \quad (25)$$

where $\lfloor x \rfloor$ is the nearest integer to x . Input the LLR vector $\gamma = (\gamma_1, \dots, \gamma_n)$ to SPA decoder of LDPC codes and consider $\hat{\mathbf{c}}$ as the output of this decoder. Convert $\hat{\mathbf{c}}$ to ± 1 notation and call the obtained vector $\hat{\mathbf{c}}'$. Estimate $\hat{\mathbf{z}}$ as follows

$$\hat{\mathbf{z}} = \left\lfloor \frac{\mathbf{y}}{4} - \frac{\hat{\mathbf{c}}'}{4} \right\rfloor. \quad (26)$$

Then, $\hat{\mathbf{x}} = \hat{\mathbf{c}}' + 4\hat{\mathbf{z}}$ is the final decoded lattice vector.

The complexity of this decoding algorithm is significantly lower than other lattices with practical decoding algorithm like LDA lattices [22] and LDLCs [20]. The decoding algorithm of LDLCs first proposed in [20] which has complexity $O(n \cdot d \cdot t \cdot \frac{1}{\Delta} \cdot \log_2(\frac{1}{\Delta}))$, where Δ is the resolution and its typical value is $1/256$, n is the dimension of lattice, t is the number of iterations and d is the average code degree. Then, in [31], a new algorithm proposed with lower complexity $O(n \cdot d \cdot t \cdot K \cdot M^3)$, where K is the number of replications, n , t and d are similar to above. Proposed typical value for K is 3 and for M it is 2 or 6. The computational complexity of the proposed decoding algorithm in [33] is only $O(n \cdot d \cdot t)$, which is significantly lower than the complexity of LDLCs' decoding.

The main purpose of this section, is constructing practically implementable lattice constellations. As we mentioned in previous section, definition of lattice constellations entails finding the intersection of shaping region and a translated infinite lattice. Thus, translating by $(-1, \dots, -1)$ in the proposed structure has no inconsistency with the definition of lattice codes. In the next section, we present efficient shaping methods to obtain a family of lattice codes based on LDPC lattices.

V. SHAPING METHODS FOR LDPC LATTICES

A. Hypercube Shaping Method

In practical channels, there exists a power constraint which is needed to be fulfilled. This entails selecting a group of lattice points with a bounded norm. In theoretical approaches, the coding lattice is intersected with a spherical shaping region to produce an efficient, power-constrained lattice code. However, spherical shaping has high computational complexity both for encoding and decoding. In [36], several efficient and practical shaping algorithms proposed for LDLC lattices. Here, we present an efficient and practical shaping algorithm for LDPC lattices based on hypercube shaping method.

In order to perform shaping, restrict the integer vector \mathbf{b} to the following finite constellation

$$b_i \in \mathcal{L}_i = \left\{ \frac{-L_i}{2}, \dots, \frac{L_i}{2} - 1 \right\}, \quad i = 1, \dots, n. \quad (27)$$

Shape the lattice codeword $\mathbf{x} = \mathbf{b}\mathbf{G}_\Lambda$ by translating each b_i by an integer multiple of L_i , $i = 1, \dots, n$. This is equivalent to transmitting a lattice point \mathbf{x}' as follows

$$\mathbf{x}' = (\mathbf{b} - \mathbf{s}\mathbf{L})\mathbf{G}_\Lambda = \mathbf{x} - \mathbf{s}\mathbf{L}\mathbf{G}_\Lambda, \quad (28)$$

where $\mathbf{L} = \text{diag}(L_1, \dots, L_n)$ is a diagonal matrix and the new integer vector is $\mathbf{b}' = \mathbf{b} - \mathbf{s}\mathbf{L}$. The choice of integer vector \mathbf{s} , depends on the employed shaping method. In hypercube shaping we choose \mathbf{s} such that the new lattice codeword components are constrained to lie in the hypercube, i.e. $|x'_i| \leq L_i$, for $i = 1, \dots, n$. The authors of [36] used a triangular structure for \mathbf{H} , the inverse of their generator matrix \mathbf{G} , with unit diagonal elements. In this case, hypercube shaping is straightforward, and s_i and x_i will be found recursively. Instead, we use the generator matrix of the lattice for shaping. In our case, the generator matrix beside the triangular structure, has a simpler form which helps us to obtain the components of \mathbf{s} directly. Since we have used a

translated and scaled lattice, we shape the lattice vectors inside the hypercube around the origin of coordinate. Then, we scale them by factor 2 and translate them by the all -1 vector. To this end, we need to solve the following system of linear equations

$$(x'_1, \dots, x'_n) = (b'_1, \dots, b'_n) \begin{bmatrix} \mathbf{I}_k & \mathbf{P}_{k \times (n-k)} \\ \mathbf{0}_{(n-k) \times k} & 2\mathbf{I}_{n-k} \end{bmatrix}, \quad (29)$$

by choosing an integer vector $\mathbf{s} = (s_1, \dots, s_n)$ such that $|x'_i| \leq L_i$ for $i = 1, \dots, n$. From (29), we have the following equations

$$x'_i = \begin{cases} b_i - L_i s_i, & i = 1, \dots, k \\ 2(b_i - L_i s_i) + \sum_{j=1}^k P_{j,i} b'_j, & i = k+1, \dots, n. \end{cases} \quad (30)$$

By choosing $s_1 = s_2 = \dots = s_k = 0$, we have $b'_i = b_i$ and $|x'_i| = |b_i| \leq L_i$, for $i = 1, \dots, k$. For $i = k+1, \dots, n$, we have the following inequalities

$$-L_i \leq 2b_i - 2L_i s_i + \sum_{j=1}^k P_{j,i} b_j \leq L_i, \quad (31)$$

or equivalently

$$\frac{b_i}{L_i} - \frac{1}{2} + \frac{1}{2L_i} \sum_{j=1}^k P_{j,i} b_j \leq s_i \leq \frac{b_i}{L_i} + \frac{1}{2} + \frac{1}{2L_i} \sum_{j=1}^k P_{j,i} b_j.$$

This interval contains only one integer number which is the unique solution

$$s_i = \left\lfloor \frac{1}{L_i} \left(b_i + \frac{1}{2} \sum_{j=1}^k P_{j,i} b_j \right) \right\rfloor. \quad (32)$$

Note that, after finding the shaped lattice codeword as discussed above, we must scale it by factor 2 and then translate it by $(-1, \dots, -1)$. Then, the shaped vectors of $2\Lambda - (1, \dots, 1)$ will be uniformly distributed over hypercube $2\mathcal{L} - (1, \dots, 1)$, where

$$\mathcal{L} = \left\{ \mathbf{x} \in \mathbb{Z}^n \left| \begin{array}{ll} -\frac{L_i}{2} \leq x_i \leq \frac{L_i}{2} - 1, & i = 1, \dots, k \\ -L_i \leq x_i \leq L_i, & i = k+1, \dots, n \end{array} \right. \right\}.$$

Since, the lattice 2Λ and shaping region \mathcal{L} are translated by the same vector and $\text{vol}(2\Lambda) = 4^{n-k} 2^k$, based on (18), the rate of our scheme is

$$R = \frac{1}{n} \log_2 \left(\frac{\text{vol}(\mathcal{L})}{\text{vol}(\Lambda)} \right) = \sum_{i=1}^k \frac{\log_2(L_i)}{n} + \sum_{i=k+1}^n \frac{\log_2\left(\frac{4L_i+2}{4}\right)}{n}. \quad (33)$$

Algorithm 1 explains the method of obtaining original information \mathbf{b} from shaped lattice codeword \mathbf{x}' . The complexity of this algorithm is $O(nd)$, where d is the average number of nonzero elements in a row of \mathbf{G}_Λ .

Algorithm 1 Obtaining original information

```

1: procedure MOD( $\mathbf{x}'$ ,  $(L_1, \dots, L_n)$ ,  $\mathbf{G}_\Lambda^{-1}$ )
2:    $\mathbf{b}' \leftarrow \left\lfloor \left( \frac{\mathbf{x}' + 1}{2} \right) \mathbf{G}_\Lambda^{-1} \right\rfloor$ 
3:   for  $i = 1 : n$  do
4:     if  $b'_i \pmod{L_i} < \frac{L_i}{2}$  then
5:        $r_i \leftarrow b'_i \pmod{L_i}$ .
6:     else
7:        $r_i \leftarrow b'_i \pmod{L_i} - L_i$ 
8:     end if
9:   end for
10:  return  $\mathbf{b} = (r_1, \dots, r_n)$ .
11: end procedure

```

B. Nested Lattice Shaping Method

Despite of its low complexity nature, hypercube shaping suffers a performance loss of 1.53dB in high dimensions compared to optimal hypersphere shaping [30]. Thus, we consider nested lattice shaping, which is suboptimal but it offers more shaping gains comparing to hypercube shaping [36]. First, limit the rate of the code by restricting the integer row vector \mathbf{b} to take values from a finite constellation in which $b_i \in \mathcal{L}_i = \{0, \dots, L_i - 1\}$ for each $i = 1, \dots, n$. Similar to the hypercube shaping, let $\mathbf{x}' = (\mathbf{b} - \mathbf{s}\mathbf{L})\mathbf{G}_\Lambda$. In this case, we choose the vector \mathbf{s} as follows

$$\mathbf{s} = \underset{\mathbf{s}_0 \in \mathbb{Z}^n}{\operatorname{argmin}} \|\mathbf{b} - \mathbf{s}_0\mathbf{L}\mathbf{G}_\Lambda\|^2. \quad (34)$$

Choosing \mathbf{s} that minimizes $\|\mathbf{x}'\|$ is equivalent to finding the nearest lattice point of the scaled lattice $\mathbf{L}\mathbf{G}$ to the non-shaped lattice point \mathbf{x} . Therefore, the codewords will be uniformly distributed along the Voronoi cell of the coarse lattice $\mathbf{L}\mathbf{G}$. Thus, the rate of the code is

$$R = \frac{1}{n} \log_2 \left(\frac{\operatorname{vol}(\mathbf{L}\Lambda)}{\operatorname{vol}(\Lambda)} \right) = \frac{\sum_{i=1}^n \log_2(L_i)}{n}. \quad (35)$$

The complexity of solving (34) is exponential in the dimension of lattice, even by restricting the components of \mathbf{b} . Using the triangular structure of the parity check matrix \mathbf{H} , the authors of [36] suggested a tree search with affordable complexity for shaping their lattices. Practically, their tree search can be done with simple sub-optimal sequential decoders such as the M -algorithm [37].

Following their method, we present a nested lattice shaping by using the triangular structure of our generator matrix.

Our algorithm starts from the first row of \mathbf{G}_Λ , and sequentially goes down along the rows. The input at row i is a list of up to M candidate sequences $\{\mathbf{s}_1^{i-1}, \dots, \mathbf{s}_M^{i-1}\}$ for $\mathbf{s}_{i-1} = (s_1, \dots, s_{i-1})$. For $i = 1$ the list is initialized with a single empty sequence. Each of these M sequences will be extended to (\mathbf{s}_j^{i-1}, s') , for $j = 1, \dots, M$, where s' takes all possible values for s_i . The algorithm assigns to each extended sequence a respective score $\sum_{j=1}^i |x'_j|^2$. The scores are sorted, and the M sequences with smallest score are retained as input to the next row. Finally, we choose \mathbf{s} as the sequence with smallest score after processing of the last row $i = n$. The storage and the computational complexity of this algorithm are determined by the value of M , as $O(ndM)$. Using an M -algorithm with $M = 1$, nested lattice shaping is equivalent to hypercube shaping, where for $M = \infty$, the algorithm is a full exponential tree search that finds the exact solution for \mathbf{s} . The shaping gain and shaping loss of LDPC lattice codes are presented in TABLE I. The results are obtained by using Monte Carlo simulation and $L_1 = L_2 = \dots = L_n = L$ assumed to be 4. An M -algorithm with $M = 5$ has been applied for nested lattice shaping. The results show that increasing the rate of the code, used as underlying code of the lattice, declines the shaping gain for hypercube shaping but it improves the shaping gain for nested lattice shaping. Increasing the dimension of the lattice also improves the shaping gain for both of the applied shaping methods. Thus, the reasonable shaping gains can be obtained by applying nested lattice shaping for high dimensional LDPC lattices, which have high rate LDPC codes as their underlying codes.

VI. ONE-WAY RELAY NETWORK

In this section, we construct a lattice-coding scheme base on LDPC lattice codes that achieves the decode-and-forward bound of (3). We present our scheme by following the scheme of [26] for LDLCs. All steps of this scheme are rephrased due to the inherent differences between LDPC lattice codes and LDLCs.

A. Decomposition of LDPC Lattice Codebooks

Based on block Markov encoding and decomposition of the full lattice codebook into lower-rate codebooks, we can propose an encoding/decoding scheme for implementing LDPC lattice

TABLE I. THE SHAPING GAIN AND SHAPING LOSS FOR LDPC LATTICE CODES IN SMALL DIMENSIONS.

(n, k)	L	Hypercube shaping		Nested lattice shaping	
		Shaping gain	Shaping loss	Shaping gain	Shaping loss
(100, 70)	4	-1.3065	2.6756	-1.0162	2.3853
(100, 80)	4	-1.4842	2.8533	-0.7846	2.1536
(100, 90)	4	-1.6665	3.0356	-0.3717	1.7408
(200, 100)	4	-0.9291	2.3653	-0.8750	2.3112
(200, 120)	4	-1.1124	2.5486	-0.8926	2.3288
(200, 140)	4	-1.2968	2.7330	-0.8395	2.2757
(200, 160)	4	-1.4798	2.9160	-0.6593	2.0955
(200, 180)	4	-1.6621	3.0983	-0.3203	1.7564
(200, 190)	4	-1.7565	3.1927	-0.0633	1.4995

codes on one-way relay channels. Let $\mathbf{b} \in \mathbb{Z}^n$ be the information vector, where each element b_i is drawn from the finite constellation of (27). Then, we define the i^{th} element of the *resolution component* \mathbf{b}_r as

$$b_{r,i} = \begin{cases} b_i, & i \in \mathcal{X} \\ 0, & i \in \{1, \dots, n\} \setminus \mathcal{X}, \end{cases} \quad (36)$$

where \mathcal{X} is a k -element subset of $\{1, \dots, n\}$, whose members are chosen randomly. Let $\mathbf{b}_v \in \mathbb{Z}^n$ be the *vestigial component* vector. Define the i^{th} element of \mathbf{b}_v as follows

$$b_{v,i} = \begin{cases} 0, & i \in \mathcal{X} \\ b_i, & i \in \{1, \dots, n\} \setminus \mathcal{X}. \end{cases} \quad (37)$$

Based on (36) and (37), define the resolution lattice codeword $\mathbf{x}_r = \mathcal{E}(b_r)$ and the vestigial lattice codeword $\mathbf{x}_v = \mathcal{E}(b_v)$, where \mathcal{E} is defined by (23). It can be easily checked that the original lattice codeword $\mathbf{x} = \mathcal{E}(\mathbf{b})$ is the sum of its resolution and vestigial components, that is

$$\mathbf{x} = \mathbf{x}_r \oplus \mathbf{x}_v = \mathbf{x}_r + \mathbf{x}_v + (1, \dots, 1). \quad (38)$$

B. Power-Constrained Decomposition of LDPC Lattices

In preceding subsection, we used unconstrained powers for codewords \mathbf{x} , \mathbf{x}_r and \mathbf{x}_v , due to the fact that we still have not enforced shaping for the lattice. Linear decomposition is straightforward

in the unconstrained power situation, but it is not trivial with power constrained scenario. We use the altered version of the proposed method in [26] with the proposed hypercube shaping method of Section IV. The steps of this method are as follows:

- 1) Map the information integer vector \mathbf{b} to \mathbf{b}' such that the i^{th} element of \mathbf{b}' is $b'_i = b_i - s_i L_i$ and

$$s_i = \begin{cases} \left\lfloor \frac{1}{L_i} \left(b_i + \frac{1}{2} \sum_{j=1}^k P_{j,i} b_j \right) \right\rfloor, & i = k+1, \dots, n \\ 0, & i = 1, \dots, k. \end{cases} \quad (39)$$

The power-constrained lattice codeword is then given as $\mathbf{x}' = \mathcal{E}(\mathbf{b}')$.

- 2) Decompose the original integer information vector \mathbf{b} as in (36) and (37).
- 3) Map the resolution component \mathbf{b}_r to \mathbf{b}'_r such that the new integer vector results in a power-constrained codeword:

$$b'_{r,i} = b_{r,i} - s_{r,i} L_i = \begin{cases} b_i - s_{r,i} L_i, & i \in \mathcal{X}, \\ -s_{r,i} L_i, & i \notin \mathcal{X}, \end{cases} \quad (40)$$

where $b_{r,i}$ and $b'_{r,i}$ are the i^{th} elements of \mathbf{b}_r and \mathbf{b}'_r , respectively. For hypercube shaping, $s_{r,i}$ can be written as,

$$s_{r,i} = \begin{cases} \left\lfloor \frac{1}{L_i} \left(b_i + \frac{1}{2} \sum_{j=1}^k P_{j,i} b_{r,j} \right) \right\rfloor, & i \in \mathcal{I} \cap \mathcal{X} \\ \left\lfloor \frac{1}{2L_i} \left(\sum_{j=1}^k P_{j,i} b_{r,j} \right) \right\rfloor, & i \in \mathcal{I} \cap \mathcal{X}^c \\ 0, & i = 1, \dots, k, \end{cases} \quad (41)$$

where $\mathcal{I} = \{k+1, \dots, n\}$. Then, the mapped lattice codeword is $\mathbf{x}'_r = \mathcal{E}(\mathbf{b}'_r)$.

- 4) For the sake of preserving the linearity of the lattice decomposition, we map the vestigial information vector \mathbf{b}_v to \mathbf{b}'_v such that the i^{th} element of \mathbf{b}'_v is given as,

$$\begin{aligned} b'_{v,i} &= b'_i - b'_{r,i} = (b_i - b_{r,i}) + (s_{r,i} - s_i) L_i \\ &= \begin{cases} (s_{r,i} - s_i) L_i, & i \in \mathcal{X} \\ b_i + (s_{r,i} - s_i) L_i, & i \notin \mathcal{X}, \end{cases} \end{aligned} \quad (42)$$

where s_i and $s_{r,i}$ are given in (39) and (41), respectively. Then, the vestigial codeword \mathbf{x}'_v can be obtained as $\mathbf{x}'_v = \mathcal{E}(\mathbf{b}'_v)$.

The primary information integers b_i , $b_{r,i}$ and $b_{v,i}$ can be recovered from b'_i , $b'_{r,i}$ and $b'_{v,i}$ by modulo L_i operation. It must be noted that the lattice codeword \mathbf{x}' and the resolution lattice

codeword \mathbf{x}'_r respect the power constraint but, the vestigial lattice codeword \mathbf{x}'_v may not fulfil the power constraint in general. However, this is not a problem for the considered relay network since we do not transmit the vestigial information alone. In order to obtain further shaping gain, we employ nested lattice shaping in our decomposition. In this case, we find the vectors \mathbf{s} and \mathbf{s}_r such that

$$\mathbf{s} = \underset{\mathbf{s}^* \in \mathbb{Z}^n}{\operatorname{argmin}} \|(\mathbf{b} - \mathbf{s}^* \mathbf{L}) \mathbf{G}_\Lambda\|^2, \quad (43)$$

$$\mathbf{s}_r = \underset{\mathbf{s}_r^* \in \mathbb{Z}^n}{\operatorname{argmin}} \|(\mathbf{b}_r - \mathbf{s}_r^* \mathbf{L}) \mathbf{G}_\Lambda\|^2. \quad (44)$$

After finding \mathbf{s} and \mathbf{s}_r , the vestigial codeword can be found as $\mathbf{x}'_v = \mathcal{E}(\mathbf{b}_v + (\mathbf{s} - \mathbf{s}_r) \mathbf{L})$.

C. Encoding and Decoding

Here, we explain the implementation of the encoding and the decoding of LDPC lattice codes, for one-way relay network. Let $\mathbf{x}'[t]$ be the full-rate power-constrained lattice codeword associated with t^{th} block, and let $\mathbf{x}'_r[t-1]$ be the resolution codeword decoded at the relay from the $[t-1]^{\text{th}}$ block. The source and the relay transmit $\sqrt{P_S} \mathbf{x}'[t]$ and $\sqrt{P_R} \mathbf{x}'_r[t-1]$ during the t^{th} block. At block $t = 1$, the source sends $\sqrt{P_S} \mathbf{x}'[1]$ and the relay sends nothing. At blocks $t = 2, \dots, T$, destination receives the superposition of signals from the source and the relay at blocks t and $t-1$, respectively. At block $t = T+1$, destination only receives the resolution information from the relay since the source has no information to send. The source transmits nTR symbols over $n(T+1)$ channel use, so the encoding rate is close to R for large T . More detail about implementing block Markov encoding using lattices can be found in [25], [26] and [28].

Following the outline that proposed in [28, Theorem 1], our decoding contains three stages. At the first stage, the relay decodes \mathbf{x}' , in the next stage, the destination decodes \mathbf{x}'_r , finally, the destination uses \mathbf{x}'_r to decode \mathbf{x}'_v . First, we consider the decoding at the relay. The received signal by the relay at the t^{th} block is

$$\mathbf{y}_R[t] = h_{SR} \sqrt{P_S} \mathbf{x}'[t] + \mathbf{z}_R[t]. \quad (45)$$

We start by scaling $\mathbf{y}_R[t]$ by the factor $1/\sqrt{P_S} h_{SR}$. Then, we use iterative decoder of LDPC lattices to obtain full-rate lattice codeword $\hat{\mathbf{x}}'[t]$

$$\hat{\mathbf{x}}'[t] = \text{DEC}_{\text{LDPC}} \left(\frac{\mathbf{y}_R[t]}{\sqrt{P_S} h_{SR}}, \frac{N_R}{\sqrt{P_S} h_{SR}} \right), \quad (46)$$

where $\text{DEC}_{\text{LDPC}}(\mathbf{y}, \sigma^2)$ is the iterative decoding algorithm of LDPC lattices described in Section IV. We use \mathbf{y} and σ^2 to estimate LLR vector in (25). The resulting information vector is denoted by $\hat{\mathbf{b}}[t]$ and can be obtained as $\hat{\mathbf{b}}[t] = \text{MOD}(\hat{\mathbf{x}}'[t], \mathbf{L}, \mathbf{G}_\Lambda^{-1})$, where \mathbf{G}_Λ^{-1} is

$$\mathbf{G}_\Lambda^{-1} = \begin{bmatrix} \frac{1}{2}\mathbf{I}_k & -\frac{1}{4}\mathbf{P}_{k \times (n-k)} \\ \mathbf{0}_{(n-k) \times k} & \frac{1}{4}\mathbf{I}_{n-k} \end{bmatrix}. \quad (47)$$

The relay estimates the resolution information $\hat{\mathbf{b}}_r[t]$ from $\hat{\mathbf{b}}[t]$ and (36), from which it finds the estimation of shaped resolution codeword $\hat{\mathbf{x}}'_r[t]$. In this case, the decoding error is $\mathbf{e}_r[t] = \mathbf{x}'_r[t] - \hat{\mathbf{x}}'_r[t]$. The received signal at the destination during $(t+1)^{\text{th}}$ block is

$$\mathbf{y}_D[t+1] = h_{RD}\sqrt{P_R}\hat{\mathbf{x}}'_r[t] + h_{SD}\sqrt{P_S}\mathbf{x}'[t+1] + \mathbf{z}_D[t+1].$$

Replace $\hat{\mathbf{x}}'_r[t]$ by $\mathbf{x}'_r[t] - \mathbf{e}_r[t]$ in $\mathbf{y}_D[t+1]$. In the destination, treat $h_{SD}\sqrt{P_S}\mathbf{x}'[t+1] + \mathbf{z}_D[t+1] - h_{RD}\sqrt{P_R}\mathbf{e}_r[t]$ as noise and decode the resolution information vector $\tilde{\mathbf{x}}'_r[t]$

$$\tilde{\mathbf{b}}'_r[t] = \left\lfloor \mathcal{D} \left(\text{DEC}_{\text{LDPC}} \left(\frac{\mathbf{y}_D[t+1]}{\sqrt{P_R}h_{RD}}, \frac{N_D}{\sqrt{P_R}h_{RD}} \right) \right) \mathbf{G}_\Lambda^{-1} \right\rfloor,$$

where $\mathcal{D}(x_1, \dots, x_n) = 0.5 \times (x_1 + 1, \dots, x_n + 1)$.

Then, the relay obtains $\tilde{\mathbf{b}}_r[t] = \text{MOD}(\tilde{\mathbf{x}}'_r[t], \mathbf{L}, \mathbf{G}_\Lambda^{-1})$, from which it finds the unshaped resolution codeword $\tilde{\mathbf{x}}_r[t]$. The decoding errors are

$$\begin{aligned} \mathbf{e}'_{d_1}[t] &= \mathbf{x}'_r[t] - \tilde{\mathbf{x}}'_r[t], \\ \mathbf{e}_{d_1}[t] &= \mathbf{x}_r[t] - \tilde{\mathbf{x}}_r[t]. \end{aligned} \quad (48)$$

Now, the destination knows the unshaped resolution codeword $\tilde{\mathbf{x}}_r[t]$ and the shaped resolution codeword $\tilde{\mathbf{x}}'_r[t]$. Next, the receiver turns to $\mathbf{y}_D[t]$, which can be written as

$$\begin{aligned} \mathbf{y}_D[t] &= h_{RD}\sqrt{P_R}(\mathbf{x}'_r[t-1] - \mathbf{e}_r[t-1]) + \\ &\quad h_{SD}\sqrt{P_S}\mathbf{x}'[t] + \mathbf{z}_D[t]. \end{aligned} \quad (49)$$

From $\mathbf{y}_D[t]$, we know $\tilde{\mathbf{x}}'_r[t-1]$. By using the linearity property (38), we can rewrite (49) as follows

$$\begin{aligned} \mathbf{y}_D[t] &= h_{RD}\sqrt{P_R}(\mathbf{x}'_r[t-1] - \mathbf{e}_r[t-1]) + \mathbf{z}_D[t] + \\ &\quad h_{SD}\sqrt{P_S}(\mathbf{x}'_r[t] + \mathbf{x}'_v[t] + (1, \dots, 1)). \end{aligned} \quad (50)$$

Now, we subtract the decoded resolution information $h_{SD}\sqrt{P_S}(\tilde{\mathbf{x}}'_r[t] + (1, \dots, 1)) + h_{RD}\sqrt{P_R}\tilde{\mathbf{x}}'_r[t-1]$ from (50) to obtain

$$\mathbf{y}'_D[t] = h_{SD}\sqrt{P_S}\mathbf{x}'_v[t] + \mathbf{e}_{d_2}[t] + \mathbf{z}_D[t], \quad (51)$$

where

$$\mathbf{e}_{d_2}[t] = h_{RD}\sqrt{P_R}(\mathbf{e}'_{d_1}[t-1] - \mathbf{e}_r[t-1]) + h_{SD}\sqrt{P_S}\mathbf{e}'_{d_1}[t].$$

Then, we use $\mathbf{y}'_D[t]$ in (51) to decode the vestigial information as follows

$$\tilde{\mathbf{x}}'_v[t] = \text{DEC}_{\text{LDPC}} \left(\frac{y'_D[t]}{\sqrt{P_S}h_{SD}}, \frac{N_D}{\sqrt{P_S}h_{SD}} \right). \quad (52)$$

Once we have decoded both the resolution and vestigial lattice codeword, the destination can find the desired lattice codeword by $\tilde{\mathbf{x}}'[t] = \tilde{\mathbf{x}}'_v[t] \oplus \tilde{\mathbf{x}}'_r[t] = \tilde{\mathbf{x}}'_v[t] + \tilde{\mathbf{x}}'_r[t] + (1, \dots, 1)$. Then, the desired integer vector can be obtained as $\tilde{\mathbf{b}}[t] = \text{MOD}(\tilde{\mathbf{x}}'[t], \mathbf{L}, \mathbf{G}_\Lambda^{-1})$.

VII. TWO-WAY RELAY NETWORKS

In this section, we present a practical block Markov encoding scheme for the two-way relay channel based on LDPC lattice codes. This scheme achieves the decode-and-forward bound in (6) and it is proposed in [28] for LDLCs. Due to the structural differences between LDPC lattices and LDLCs, as well as, differences between our shaping methods and theirs, all steps of this scheme are rephrased for LDPC lattice codes.

Our applied scheme relies on doubly-nested lattice codes in which the two sources use different codebooks with different transmit powers. Let $\Lambda_{S_1}^s \subset \Lambda_{S_1}^c$ and $\Lambda_{S_2}^s \subset \Lambda_{S_2}^c$ be the shaping and coding lattices for S_1 and S_2 , respectively. Consider the codebooks $\mathcal{C}_{S_1} = \Lambda_{S_1}^c \cap \mathcal{V}(\Lambda_{S_1}^s)$ and $\mathcal{C}_{S_2} = \Lambda_{S_2}^c \cap \mathcal{V}(\Lambda_{S_2}^s)$. It is also assumed by the authors of [28] that $h_{S_1R}P_{S_1} \geq h_{S_2R}P_{S_2}$, $\sigma^2(\Lambda_{S_1}^s) = h_{S_1R}P_{S_1}$, and $\sigma^2(\Lambda_{S_2}^s) = h_{S_2R}P_{S_2}$. There is another lattice, which is referred to as *meso-lattice*, that partitions the lattice codebook into lower-rate constituent codebooks [28]. The meso-lattice Λ_m , is nested according to $\Lambda_{S_1}^s \subset \Lambda_{S_2}^s \subset \Lambda_m \subset \Lambda_{S_1}^c, \Lambda_{S_2}^c$, where $\Lambda_{S_1}^c$ and $\Lambda_{S_2}^c$ are also nested lattices. Define the resolution codebooks $\mathcal{C}_{S_1}^{(r)} = \Lambda_{S_1}^c \cap \mathcal{V}(\Lambda_m)$ and $\mathcal{C}_{S_2}^{(r)} = \Lambda_{S_2}^c \cap \mathcal{V}(\Lambda_m)$. Then, the vestigial codebooks are $\mathcal{C}_{S_1}^{(v)} = \Lambda_m \cap \mathcal{V}(\Lambda_{S_1}^s)$ and $\mathcal{C}_{S_2}^{(v)} = \Lambda_m \cap \mathcal{V}(\Lambda_{S_2}^s)$. Let $R_{S_1}, R_{S_2}, R_{S_1}^{(r)}, R_{S_2}^{(r)}, R_{S_1}^{(v)}, R_{S_2}^{(v)}$ be the rates of $\mathcal{C}_{S_1}, \mathcal{C}_{S_2}, \mathcal{C}_{S_1}^{(r)}, \mathcal{C}_{S_2}^{(r)}, \mathcal{C}_{S_1}^{(v)}, \mathcal{C}_{S_2}^{(v)}$, respectively. Then, as before, $R_{S_1} = R_{S_1}^{(r)} + R_{S_1}^{(v)}$ and $R_{S_2} = R_{S_2}^{(r)} + R_{S_2}^{(v)}$. Thus, each full-rate codeword can be decomposed into a unique modulo sum of resolution and vestigial codewords. It is proved that the preceding code construction achieves near-capacity rates [28].

A. Decomposition of LDPC Lattice Codebooks

Let $\mathbf{b} \in \mathbb{Z}^n$ be the information vector, where each element b_i of \mathbf{b} is drawn from finite constellations $\{-L_i/2, \dots, (L_i/2) - 1\}$ and $\{0, \dots, L_i - 1\}$ for hypercube shaping and nested lattice shaping, respectively. The i^{th} element of the resolution component $\mathbf{b}^{(r)}$ is

$$b_i^{(r)} = b_i \pmod{L_i^{(r)}}, \quad (53)$$

where $L_i = \beta L_i^{(r)}$, for some $\beta \in \mathbb{Z}$. Thus, the i^{th} element of $\mathbf{b}^{(r)}$ lies in the finite constellation $\mathcal{L}_i^{(r)} = \{0, \dots, L_i^{(r)} - 1\}$ and the rate of the obtained codebook $R^{(r)}$ can be acquired via (33) or (35), depending on the applied shaping method, by replacing L_i 's with $L_i^{(r)}$'s. The vestigial component is defined as follows

$$\mathbf{b}^{(v)} = \mathbf{b} - \mathbf{b}^{(r)}. \quad (54)$$

If the employed shaping method is hypercube shaping, it can be shown that the i^{th} element of $\mathbf{b}^{(v)}$ lies in the following finite constellation

$$\mathcal{L}_i^{(v)} = \left\{ \frac{-\beta}{2} L_i^{(r)}, \frac{(-\beta+2)}{2} L_i^{(r)}, \dots, \frac{(\beta-2)}{2} L_i^{(r)} \right\}. \quad (55)$$

If the employed shaping method is nested lattice shaping, the i^{th} element of $\mathbf{b}^{(v)}$ lies in the finite constellation $\{0, L_i^{(r)}, 2L_i^{(r)}, \dots, (\beta-1)L_i^{(r)}\}$. The rate of the vestigial codebook is approximately $R^{(v)} = \log_2(\beta)$. Due to the aforementioned assumptions, each codeword $\mathbf{x} = \mathcal{E}(\mathbf{b})$ decomposes into its resolution component $\mathbf{x}^{(r)} = \mathcal{E}(\mathbf{b}^{(r)})$ and its vestigial component $\mathbf{x}^{(v)} = \mathcal{E}(\mathbf{b}^{(v)})$ as $\mathbf{x} = \mathbf{x}^{(r)} \oplus \mathbf{x}^{(v)} = \mathbf{x}^{(r)} + \mathbf{x}^{(v)} + (1, \dots, 1)$.

B. Power-Constrained Decomposition of LDPC Lattice Codebooks

The first step in shaping the lattice codeword \mathbf{x} is translating b_i by an integer multiple of L_i . Then, following the same procedure as in Section V, the shaped lattice point is obtained. The shaped lattice codeword is $\mathbf{x}' = \mathcal{E}(\mathbf{b} - \mathbf{s}\mathbf{L})$, where \mathbf{s} is given in (32) and (34), for hypercube shaping and nested lattice shaping, respectively. Next, we adapt the shaping methods to the resolution codewords in such a way that the decomposition of the lattice codebook remains linear. We shape the resolution component $\mathbf{b}^{(r)}$ to $\mathbf{b}'^{(r)}$ as follows

$$b_i'^{(r)} = \text{smod}(b_i, L_i^{(r)}) - s_i^{(r)} L_i^{(r)}, \quad (56)$$

where b_i and $b_i^{(r)}$ are the i^{th} elements of \mathbf{b} and $\mathbf{b}^{(r)}$, respectively, and

$$\text{smod}(x, L) = \begin{cases} \bar{x} = x \pmod{L}, & \text{if } \bar{x} < \frac{L}{2}, \\ x \pmod{L} - L, & \text{otherwise.} \end{cases} \quad (57)$$

We choose the elements of $\mathbf{s}^{(r)}$ according to

$$s_i^{(r)} = \left\lfloor \frac{1}{L_i^{(r)}} \left(b_i^{(r)} + \frac{1}{2} \sum_{j=1}^k P_{j,i} b_j^{(r)} \right) \right\rfloor, \quad i = 1, \dots, n, \quad (58)$$

for hypercube shaping. For nested lattice shaping, we consider $b_i^{(r)} = b_i \pmod{L_i^{(r)}} - s_i^{(r)} L_i^{(r)}$, where $\mathbf{s}^{(r)}$ is given by (44). Indeed, the smod function is regular modulo operation, when the employed shaping method is nested lattice shaping. Then, the shaped resolution component is given by $\mathbf{x}'^{(r)} = \mathcal{E}(\mathbf{b}'^{(r)})$. In order to preserve the modulo linearity of the lattice decomposition, we map the vestigial integer vector $\mathbf{b}^{(v)}$ to $\mathbf{b}'^{(v)}$ as follows

$$\mathbf{b}'^{(v)} = \mathbf{b}' - \mathbf{b}^{(r)} = \mathbf{b}^{(v)} - \mathbf{s}\mathbf{L}, \quad (59)$$

where \mathbf{s} is given in (32) and (34), for hypercube shaping and nested lattice shaping, respectively. According to this decomposition, we have $\mathbf{b}' \neq \mathbf{b}'^{(r)} + \mathbf{b}'^{(v)}$ in general. However, the decomposition preserves componentwise modulo linearity, that is, $\mathbf{b}'_i \pmod{L_i} = \mathbf{b}'^{(r)}_i \pmod{L_i^{(r)}} + \mathbf{b}'^{(v)}_i \pmod{L_i}$. Then, the vestigial codeword is $\mathbf{x}'^{(v)} = \mathcal{E}(\mathbf{b}'^{(v)})$. It can be easily shown that $\mathbf{x}' = \mathbf{x}^{(r)} \oplus \mathbf{x}'^{(v)}$. It should be noted that the vestigial lattice codeword \mathbf{x}'_v may not fulfil the power constraint, which is not a problem for the considered relay network, since we do not transmit the vestigial information alone. The original information vectors $\mathbf{b}, \mathbf{b}^{(v)}$ can be recovered from $\mathbf{b}', \mathbf{b}'^{(v)}$ by using Algorithm 1 with $\mathbf{L} = (L_1, \dots, L_n)$. Similarly, the resolution information vector $\mathbf{b}^{(r)}$ can be recovered from $\mathbf{b}'^{(r)}$ by using Algorithm 1 with $\mathbf{L}^{(r)} = (L_1^{(r)}, \dots, L_n^{(r)})$.

C. Encoding and Decoding

Here, we present the implementation of encoding and decoding for two-way relay network using LDPC lattice codes. First, we describe the encoding scheme and then we discuss the decoding schemes at each node. The two sources employ LDPC lattice codes as described in previous subsection. The full-rate information vector, resolution information vector, vestigial information vector, full-rate codeword, resolution codeword, and vestigial codeword of source i are denoted by $\mathbf{b}_{S_i}, \mathbf{b}_{S_i}^{(r)}, \mathbf{b}_{S_i}^{(v)}, \mathbf{x}_{S_i}, \mathbf{x}_{S_i}^{(r)}$ and $\mathbf{x}_{S_i}^{(v)}$, respectively, for $i = 1, 2$. Let the constellation

sizes of the i^{th} element of \mathbf{b}_{S_1} and \mathbf{b}_{S_2} be $L_{S_1,i}$ and $L_{S_2,i}$, respectively. Further, let $L_i^{(r)}$ be the constellation size of the resolution codeword, which is selected such that $L_i^{(r)}$ divides both $L_{S_1,i}$ and $L_{S_2,i}$ [28]. It is assumed by the authors of [28] that the sources use transmit powers such that

$$\frac{\sqrt{P_{S_1}}h_{S_1R}}{m_1} = \frac{\sqrt{P_{S_2}}h_{S_2R}}{m_2} = \rho, \quad (60)$$

where $m_1, m_2 \in \mathbb{Z}$ and $\rho \in \mathbb{R}$. Furthermore, it is assumed that $\gcd(L_i^{(r)}, m_1) = \gcd(L_i^{(r)}, m_2) = 1$. In the rest of this paper, we assume $m_1 = m_2 = 1$.

First, we describe the encoding steps. The two sources S_1 and S_2 transmit their signals to the relay and to each other. Meanwhile, the relay transmits its own signal to both S_1 and S_2 . Thus, S_1 receives the superposition of signals from S_2 and the relay. Similarly, S_2 receives signals from S_1 and the relay. During the t^{th} block, where $t = 2, \dots, T$, the sources S_1 and S_2 transmit their new codewords $\sqrt{P_{S_1}}\mathbf{x}'_{S_1}[t]$ and $\sqrt{P_{S_2}}\mathbf{x}'_{S_2}[t]$, while the relay transmits the resolution component of the decoded sum codeword $\sqrt{P_R}\mathbf{x}_R'^{(r)}[t-1]$. During the block $t = 1$, the relay transmits nothing and at block $t = T+1$, the sources receive the resolution information $\sqrt{P_R}\mathbf{x}_R'^{(r)}[T]$ from the relay, since both sources have no information to send.

Next, we describe the decoding steps. Since, the decoding at S_1 and S_2 is similar, we only consider decoding at S_2 . Decoding occurs in three phases. In phase one, the relay decodes the sum codeword $\mathbf{x}'_{S_1} \oplus \mathbf{x}'_{S_2}$. In phase two, S_2 decodes the resolution codeword $\mathbf{x}_{S_1}^{(r)}$ by treating other codewords as noise. In the last phase, S_2 decodes the vestigial codeword $\mathbf{x}_{S_1}^{(v)}$.

The received signal at relay in t^{th} block is

$$\begin{aligned} \mathbf{y}_R[t] &= \sqrt{P_{S_1}}h_{S_1R}\mathbf{x}'_{S_1}[t] + \sqrt{P_{S_2}}h_{S_2R}\mathbf{x}'_{S_2}[t] + \mathbf{z}_R[t] \\ &= \rho (m_1\mathbf{x}'_{S_1}[t] + m_2\mathbf{x}'_{S_2}[t]) + \mathbf{z}_R[t]. \end{aligned} \quad (61)$$

Let $\mathbf{y}'_R[t] = \mathbf{y}_R[t] - \rho(-m_1 - m_2 + 1)$. Using $\mathbf{y}'_R[t]$, the relay performs LDPC lattice decoding to obtain an estimate of the sum information $m_1\mathbf{b}'_{S_1}[t] + m_2\mathbf{b}'_{S_2}[t]$

$$m_1\hat{\mathbf{x}}'_{S_1}[t] \oplus m_2\hat{\mathbf{x}}'_{S_2}[t] = \text{DEC}_{\text{LDPC}}\left(\frac{\mathbf{y}'_R[t]}{\rho}, \frac{N_R}{\rho}\right), \quad (62)$$

$$m_1\hat{\mathbf{b}}'_{S_1}[t] + m_2\hat{\mathbf{b}}'_{S_2}[t] = \lfloor \mathcal{D}(\hat{\mathbf{x}}'_R[t]) \mathbf{G}_\Lambda^{-1} \rfloor, \quad (63)$$

where function \mathcal{D} is defined in Section VI-C and $\hat{\mathbf{x}}'_R[t] = m_1\hat{\mathbf{x}}'_{S_1}[t] + m_2\hat{\mathbf{x}}'_{S_2}[t]$. The i^{th} element

of the sum is given by

$$\begin{aligned}
\hat{b}'_{R,i} &= m_1 \hat{b}'_{S_1,i} + m_2 \hat{b}'_{S_2,i} \\
&= m_1 (\hat{b}_{S_1,i} - s_{1,i} L_{S_1,i}) + m_2 (\hat{b}_{S_2,i} - s_{2,i} L_{S_2,i}) \\
&= m_1 \hat{b}_{S_1,i} + m_2 \hat{b}_{S_2,i} - m_2 s_{2,i} L_{S_2,i} - m_1 s_{1,i} L_{S_1,i}.
\end{aligned} \tag{64}$$

Then, the relay takes the result modulo $L_i^{(r)}$ to find the modulo sum of the two resolution information vectors

$$\hat{b}_{R,i}^{(r)} = \text{smod} \left(\left[m_1 \hat{b}_{S_1,i} + m_2 \hat{b}_{S_2,i} \right], L_i^{(r)} \right) \tag{65}$$

$$= \text{smod} \left(\left[m_1 \hat{b}'_{S_1,i} + m_2 \hat{b}'_{S_2,i} \right], L_i^{(r)} \right). \tag{66}$$

Since $L_i^{(r)}$ and m_1, m_2 were selected to be co-primes, the individual codewords can be recovered from a single codeword and the modulo sum. Due to the existent constraints on transmit power, the relay employs shaping methods and maps $\hat{\mathbf{b}}_R^{(r)}[t]$ to another information vector $\hat{\mathbf{b}}_R'^{(r)}[t]$

$$\hat{\mathbf{b}}_R'^{(r)}[t] = \hat{\mathbf{b}}_R^{(r)}[t] - \mathbf{s}_R \mathbf{L}^{(r)}, \tag{67}$$

where \mathbf{s}_R is given in (58) and (44) for hypercube shaping and nested lattice shaping, respectively. Then, the shaped lattice codeword is $\hat{\mathbf{x}}_R'^{(r)}[t] = \mathcal{E} \left(\hat{\mathbf{b}}_R'^{(r)}[t] \right)$. During block $t+1$, the relay transmits $\sqrt{P_R} \hat{\mathbf{x}}_R'^{(r)}[t]$.

Now, we describe the decoding process at S_2 . The received signal at S_2 in block $t+1$ is

$$\begin{aligned}
\mathbf{y}'_{S_2}[t+1] &= h_{RS_2} \sqrt{P_R} \hat{\mathbf{x}}_R'^{(r)}[t] \\
&+ h_{S_1 S_2} \sqrt{P_{S_1}} \mathbf{x}'_{S_1}[t+1] + \mathbf{z}_{S_2}[t+1].
\end{aligned} \tag{68}$$

Source 2 subtracts its own scaled and translated codeword $m_2 h_{RS_2} \sqrt{P_R} [\mathbf{x}_{S_2}[t] + (1, \dots, 1)]$ from the received signal to obtain

$$\begin{aligned}
\mathbf{y}''_{S_2}[t+1] &= h_{RS_2} \sqrt{P_R} \left[\hat{\mathbf{x}}_R'^{(r)}[t] - m_2 \mathbf{x}_{S_2}[t] - (1, \dots, 1) \right] \\
&+ h_{S_1 S_2} \sqrt{P_{S_1}} \mathbf{x}'_{S_1}[t+1] + \mathbf{z}_{S_2}[t+1].
\end{aligned} \tag{69}$$

Then, treating $h_{S_1 S_2} \sqrt{P_{S_1}} \mathbf{x}'_{S_1}[t+1] + \mathbf{z}_{S_2}[t+1]$ as noise, S_2 uses LDPC lattice decoding to obtain

$$\tilde{\mathbf{b}}_{S_1}^{(r)}[t] = \left[\mathcal{D} \left(\text{DEC}_{\text{LDPC}} \left(\frac{\mathbf{y}''_{S_2}[t+1]}{\gamma}, \frac{N_{S_2}}{\gamma} \right) \right) \mathbf{G}_A^{-1} \right], \tag{70}$$

where $\gamma = h_{RS_2}\sqrt{P_R}$. It must be noted that the decoded integer vector in (70) is

$$\tilde{\mathbf{b}}_{S_1}^{(r)}[t] = \text{smod} \left(\tilde{\mathbf{b}}_R^{(r)}[t], \mathbf{L}^{(r)} \right) - m_2 \mathbf{b}_{S_2}[t] - \mathbf{s}_R \mathbf{L}^{(r)}, \quad (71)$$

where $\tilde{\mathbf{b}}_R^{(r)}[t] = m_1 \tilde{\mathbf{b}}_{S_1}[t] + m_2 \tilde{\mathbf{b}}_{S_2}[t]$. Let $\mathbf{b}_1[t] = \text{smod} \left(\left[\tilde{\mathbf{b}}_{S_1}^{(r)}[t] + m_2 \mathbf{b}_{S_2}[t] \right], \mathbf{L}^{(r)} \right)$ and $\mathbf{b}_2[t] = \text{smod2mod}(\mathbf{b}_1[t], \mathbf{L}^{(r)})$, in which the function $\text{smod2mod}(x, L)$, that is defined next, is applied componentwise

$$\text{smod2mod}(x, L) = \begin{cases} x, & \text{if } 0 \leq x \leq \frac{L}{2} - 1, \\ x + L, & \text{if } -\frac{L}{2} \leq x < 0. \end{cases} \quad (72)$$

Source 2 obtains the resolution information of source 1 as

$$\begin{aligned} (\mathbf{b}_2[t] - m_2 \mathbf{b}_{S_2}[t]) \pmod{\mathbf{L}^{(r)}} \\ &= \left[\left(m_1 \tilde{\mathbf{b}}_{S_1}[t] + m_2 \tilde{\mathbf{b}}_{S_2}[t] \right) \pmod{\mathbf{L}^{(r)}} \right. \\ &\quad \left. - m_2 \mathbf{b}_{S_2}[t] - \mathbf{s}_R \mathbf{L}^{(r)} \right] \pmod{\mathbf{L}^{(r)}} \\ &= m_1 \tilde{\mathbf{b}}_{S_1}[t] \pmod{\mathbf{L}^{(r)}}. \end{aligned} \quad (73)$$

Then, it recovers the i^{th} element of unshaped resolution information $\tilde{\mathbf{b}}_{S_1}^{(r)}[t]$ by computing

$$\tilde{b}_{S_1,i}^{(r)} = \frac{(m_1 \tilde{b}_{S_1,i}) \pmod{L_i^{(r)}} + \delta L_i^{(r)}}{m_1}, \quad (74)$$

where δ is the unique integer such that $\tilde{b}_{S_1,i}^{(r)} \in \{0, \dots, L_i^{(r)}\}$. Such a unique δ always exists since $L_i^{(r)}$ and m_1 are coprime [28]. Next, S_2 uses $\tilde{\mathbf{b}}_{S_1}^{(r)}[t]$ and $\tilde{\mathbf{b}}_{S_1}^{(r)}[t] + m_2 \mathbf{b}_{S_2}[t]$ to obtain $\tilde{\mathbf{x}}_{S_1}^{(r)}[t] = \mathcal{E}(\tilde{\mathbf{b}}_{S_1}^{(r)}[t])$ and $\tilde{\mathbf{x}}_R^{(r)}[t] = \mathcal{E}(\tilde{\mathbf{b}}_{S_1}^{(r)}[t] + m_2 \mathbf{b}_{S_2}[t])$. Hence, it has $\tilde{\mathbf{x}}_R^{(r)}[t-1]$ and $\tilde{\mathbf{x}}_{S_1}^{(r)}[t]$ from $\mathbf{y}'_{S_2}[t]$ and $\mathbf{y}'_{S_2}[t+1]$. Then, S_2 subtracts $h_{RS_2}\sqrt{P_R}\tilde{\mathbf{x}}_R^{(r)}[t-1] + h_{S_1S_2}\sqrt{P_{S_1}}\left(\tilde{\mathbf{x}}_{S_1}^{(r)}[t] + (1, \dots, 1)\right)$ from the original received signal at t^{th} block $\mathbf{y}'_{S_2}[t]$. The simplified form of this subtracted signal is

$$\mathbf{y}_{S_2}'''[t] = h_{S_1S_2}\sqrt{P_{S_1}}\mathbf{x}_{S_1}'^{(v)}[t] + \mathbf{e}_{S_2} + \mathbf{z}_{S_2}[t], \quad (75)$$

where \mathbf{e}_{S_2} is the decoding error

$$\begin{aligned} \mathbf{e}_{S_2} &= h_{S_1S_2}\sqrt{P_{S_1}} \left[\mathbf{x}_{S_1}^{(r)}[t] - \tilde{\mathbf{x}}_{S_1}^{(r)}[t] \right] \\ &\quad + h_{RS_2}\sqrt{P_R} \left[\hat{\mathbf{x}}_R^{(r)}[t-1] - \tilde{\mathbf{x}}_R^{(r)}[t-1] \right]. \end{aligned} \quad (76)$$

Then, S_2 uses LDPC lattice decoding to find

$$\tilde{\mathbf{b}}_{S_1}'^{(v)}[t] = \mathbf{L}^{(r)} \circ \left[\mathcal{D}(\text{DEC}_{\text{LDPC}}(\mathbf{y}_{S_2}, \boldsymbol{\sigma}_{S_2})) \mathbf{G}_A^{-1} \right], \quad (77)$$

where \circ denotes the Hadamard product or entrywise product of matrices, $y_{S_2,i} = \frac{y_{S_2,i}'''}{L_i^{(r)} h_{S_1 S_2} \sqrt{P_{S_1}}}$ and $\sigma_{S_2,i} = \frac{N_{S_2}}{L_i^{(r)} h_{S_1 S_2} \sqrt{P_{S_1}}}$.

To obtain the vestigial information, S_2 computes $\tilde{\mathbf{b}}_{S_1}^{(v)}[t] = \text{smod}(\tilde{\mathbf{b}}_{S_1}^{(v)}[t], \mathbf{L}_{S_1})$. Finally, S_2 obtains the shaped and unshaped full-rate information by taking the sum of the resolution and vestigial information as $\tilde{\mathbf{b}}_{S_1}'[t] = \tilde{\mathbf{b}}_{S_1}^{(r)}[t] + \tilde{\mathbf{b}}_{S_1}^{(v)}[t]$ and $\tilde{\mathbf{b}}_{S_1}[t] = \tilde{\mathbf{b}}_{S_1}^{(r)}[t] + \tilde{\mathbf{b}}_{S_1}^{(v)}[t]$, respectively. A symbol error occurs at S_2 if $\tilde{b}_{S_1,i} \neq b_{S_1,i}$.

Note that the above decoding process is presented for the case that the employed shaping method is hypercube shaping. When the employed shaping is nested lattice shaping, this decoding steps are still valid by changing smod function into regular modulo operation. This is due to the fact that, the components of lattice vectors, given as the inputs for hypercube shaping and nested lattice shaping, are drawn from different sets. For hypercube and nested lattice shading methods we use the sets $\{-L/2, \dots, L/2 - 1\}$ and $\{0, \dots, L - 1\}$, respectively.

VIII. NUMERICAL RESULTS

A. One-Way Relay Channels

In the simulations, $|\mathcal{X}| = n/2$, i.e., we assume 50% of the information integers are zero for resolution and vestigial information vectors. We have used binary LDPC codes with $(n, k) = (1000, 850), (5000, 4250)$, where n and k are the codeword length and the dimension of the code, respectively. Symbol error rate (SER) performance of LDPC lattice codes are plotted against the sum power at source and the relay, i.e., $P_S E\{x_S^2\} + P_R E\{x_R^2\}$. We have considered $d_1 = d_{SR} = 0.9$, $d_2 = d_{RD} = 0.1$ and $d_{SD} = 1$. The path loss exponents are $\alpha_1 = 1$, $\alpha_2 = 2$. The variances of the noise at the relay and destination are $N_r = N_d = 0\text{dB}$. The maximum number of iterations in each step of the decoding is assumed to be 50. Since, the encoder and the decoder both know the locations of the zeros in resolution and vestigial information, based on (39)–(42), for following locations we have

$$\begin{cases} x'_{v,i} = 2b'_{v,i} - 1 = -1, & i \in \mathcal{I}^c \cap \mathcal{X} \\ x'_{r,i} = 2b'_{r,i} - 1 = -1, & i \in \mathcal{I}^c \cap \mathcal{X}^c. \end{cases} \quad (78)$$

We estimate $E\{x_S^2\}$ and $E\{x_R^2\}$ for our scheme, for the case that hypercube shaping is applied. When, all of the elements of the lattice codewords are uniformly distributed over $(L/2, L/2)$, the average power of x_i is $E\{x_i^2\} = L_i^2/12$. Due to the fact that, the resolution lattice vectors

contain more zeros, the average power in this case is less than $L_i^2/12$. We assume that the members of incoming integer vector are uniformly distributed over $\mathcal{L}_i = \{-L_i/2, \dots, L_i/2 - 1\}$, for $i = 1, \dots, n$. Since $E\{x_i\} = -0.5$ for $i \in \mathcal{I}^c$ and $E\{x_i\} = 0$ for $i \in \mathcal{I}$, we have $E\{(2x_i - 1)^2\} = 4E\{x_i^2\} + 1$ for $i \in \mathcal{I}$, and $E\{(2x_i - 1)^2\} = 4E\{x_i^2\} + 3$ for $i \in \mathcal{I}^c$, and from above equations we have

$$E\{x_i'^2\} = \begin{cases} 4 \times \frac{L_i^2 + (L_i - 2)^2}{24} + 3, & i \in \mathcal{I}^c \\ 4 \times \frac{4L_i^2}{12} + 1, & i \in \mathcal{I}. \end{cases} \quad (79)$$

Put $L_1 = L_2 = \dots = L_n = L$, then we have

$$\begin{aligned} E\{\mathbf{x}_S^2\} &= \frac{\sum_{i=1}^n E\{x_i'^2\}}{n} \\ &= \frac{k(L^2 + (L - 2)^2 + 18) + 8(n - k)(L^2 + 6)}{6n}. \end{aligned} \quad (80)$$

For $i \in \mathcal{I}^c \cap \mathcal{X}^c$, $x_{r,i}' = -1$, so $E\{x_{r,i}'^2\} = 1$. For $i \in \mathcal{I}^c \cap \mathcal{X}$, $x_{r,i}' = 2b_i - 1$. Thus,

$$\begin{aligned} E\{\mathbf{x}_R^2\} &= \frac{|\mathcal{I}^c \cap \mathcal{X}|((1/6)(L^2 + (L - 2)^2) + 3)}{n} \\ &\quad + \frac{(n - k)(4L^2 + 1)}{3n} + \frac{|\mathcal{I}^c \cap \mathcal{X}^c|}{n}. \end{aligned} \quad (81)$$

We have considered $L_i = 8$, for $i = 1, \dots, n$. Thus, for the cases that we have employed hypercube shaping, based on (33), the corresponding rate is 3.01 bits/integer. For employing nested lattice shaping, we consider $L_1 = \dots = L_k = 8$ and $L_{k+1} = \dots = L_n = 4$. Then, based on (35), the corresponding rate is 2.85 bits/integer. In order to achieve these rates, according to (3), the total required powers are $P_1 = P_S E\{\mathbf{x}_S^2\} + P_R E\{\mathbf{x}_R^2\} \geq 51.88 = 17.15\text{dB}$, and $P_2 \geq 41.2 = 16.15\text{dB}$, respectively.

In Fig. 2, we have presented SER variation versus sum of transmit powers for both nested-lattice shaping and hypercube shaping. In [26], the implementation of block Markov encoding proposed for LDLC lattice codes. We have considered h_{SD} , h_{SR} , h_{RD} and other parameters similar to their corresponding values in [26]. The SER performance of an LDLC lattice code with dimension 1000 and rate 2.78, which is obtained by employing nested lattice shaping, at 10^{-4} is 3.77dB away from its corresponding DF inner bound, which is 15.77dB. We observe that the SER performance of an LDPC lattice code of length 1000 at 10^{-4} is 4.5dB away from its corresponding DF inner bound. This is a natural result, due to the better SER performance of LDLCs comparing to LDPC lattice codes, over AWGN channels. Different decoders have

been proposed for LDLCs. As far as we know, the best one is proposed in [31]. The decoding complexity of LDLCs, by using proposed decoder in [31], is at least 24 times more than the decoding complexity of LDPC lattices. Indeed, the decoding complexity of an LDPC lattice of dimension 1000 is equivalent to the decoding complexity of an LDLC with dimension 24000. Results of Fig.2 show that the increase in the dimension of the lattice can decrease the gap between DF bound and the performance curve. Using an LDPC lattice code of dimension 5000 instead of dimension 1000 makes about 0.55dB improvement in the performance.

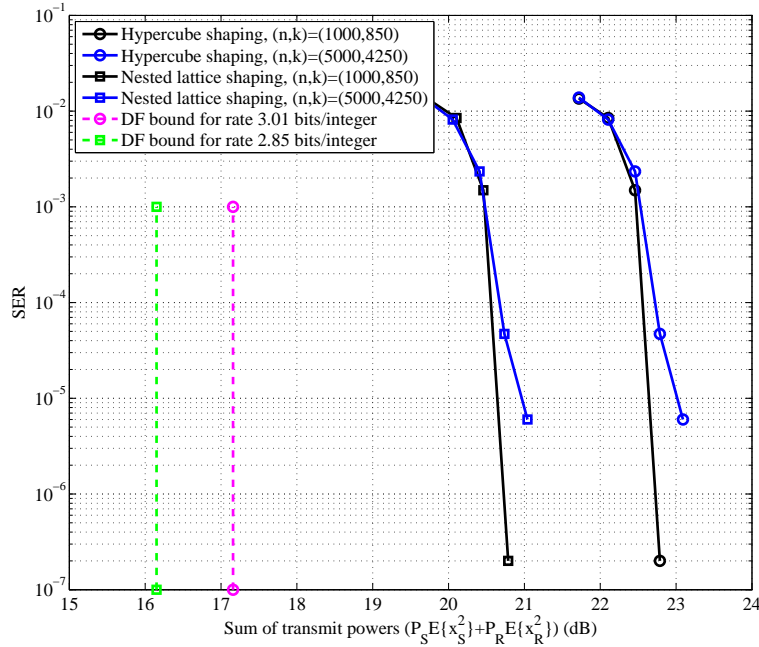


Fig. 2. Symbol error rate of LDPC lattice codes over the one-way relay channel.

B. Two-Way Relay Channels

In Fig.3, we plot the SER versus the sum of transmit powers ($\sum_{i=1}^2 P_{S_i} E\{\mathbf{x}_{S_i}^2\} + P_R E\{\mathbf{x}_R^2\}$), for the two-way relay channel. We suppose that the relay is midway between the sources, that is, $d_{S_1R} = d_{S_2R} = 0.5$ and $d_{S_1S_2} = 1$. We assume $L_{S_1,i} = L_{S_2,i} = 8$, for $i = 1, \dots, k$ and $L_{S_1,i} = L_{S_2,i} = 4$, for $i = k + 1, \dots, n$. For the resolution lattice we consider $\mathbf{L}^{(r)} = (2, \dots, 2)$. We choose $N_{S_1} = N_{S_2} = N_R = 0$ dB. Path-loss exponents are $\alpha_{S_1R} = \alpha_{RS_1} = 1$ and $\alpha_{RS_2} = \alpha_{S_2R} = 5$. The used underlying codes are the same ones that we used for one-way relay channels.

Depending on the dimension of the lattice and the applied shaping method, our scheme achieves to within 2dB of the achievable rate in (6).

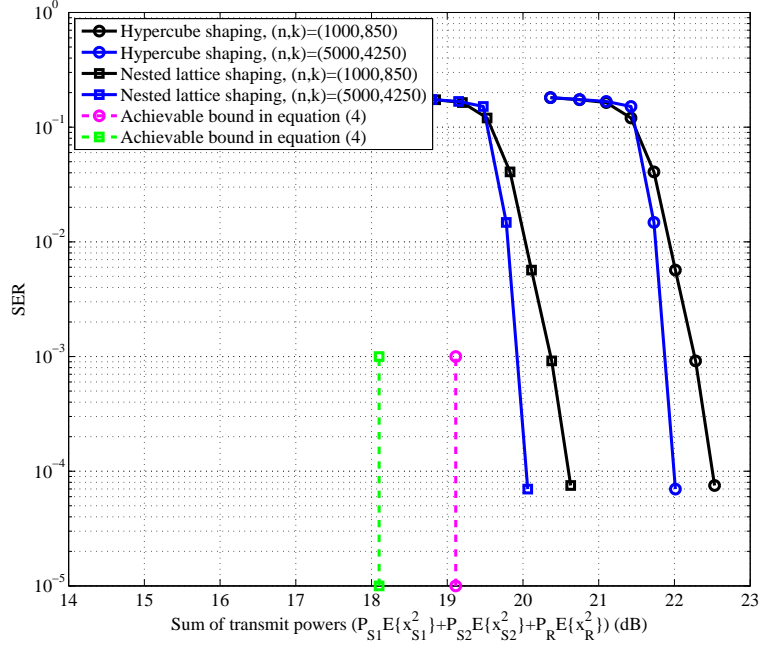


Fig. 3. Symbol error rate of LDPC lattice codes over the two-way relay channel.

IX. CONCLUSIONS

In this paper, we present the implementation of block Markov encoding using LDPC lattice codes over the one-way and two-way relay channels. In order to apply this scheme, we employ a low complexity decoding method for LDPC lattices. Then, for using these lattices in the power constrained scenarios, we propose two efficient shaping methods based on hypercube shaping and nested lattice shaping. We apply different decomposition schemes for one-way and two-way relay channels. The applied decomposition schemes are the altered versions of the applied methods for decomposing LDLCs. Due to the lower complexity of decoding LDPC lattices comparing to LDLCs, the complexity of the proposed schemes in this paper are significantly lower than the ones proposed for LDLCs. Simulation results show that LDLCs outperform LDPC lattices in general. However, having lower decoding complexity enables us to increase the dimension of the lattice to compensate the existing gap between the performance of the LDPC lattice codes and the LDLCs.

REFERENCES

- [1] H. Khodaiemehr, D. Kiani, and M.-R. Sadeghi, "One-level LDPC lattice codes for the relay channels," in *Iran Workshop on Commun. and Inform. Theory (IWCIT)*, 2015, May 2015, pp. 1–6.
- [2] E. C. Van Der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.
- [3] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. on Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.
- [4] A. E. Gamal and M. Aref, "The capacity of the semideterministic relay channel (corresp.)," *IEEE Trans. on Inform. Theory*, vol. 28, no. 3, pp. 536–536, May 1982.
- [5] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. on Inform. Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011.
- [6] S. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Trans. on Inform. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.
- [7] H.-F. Chong, M. Motani, and H. Garg, "Generalized backward decoding strategies for the relay channel," *IEEE Trans. on Inform. Theory*, vol. 53, no. 1, pp. 394–401, Jan. 2007.
- [8] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. part I. system description," *IEEE Trans. on Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.
- [9] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. on Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [10] O. Sahin, O. Simeone, and E. Erkip, "Interference channel with an out-of-band relay," *IEEE Trans. on Inform. Theory*, vol. 57, no. 5, pp. 2746–2764, May 2011.
- [11] N. Lee and C. Wang, "Aligned interference neutralization and the degrees of freedom of the two-user wireless networks with an instantaneous relay," *IEEE Trans. on Commun.*, vol. 61, no. 9, pp. 3611–3619, Sep. 2013.
- [12] I. Shomorony and A. S. Avestimehr, "Degrees of freedom of two-hop wireless networks: Everyone gets the entire cake," *IEEE Trans. on Inform. Theory*, vol. 60, no. 5, pp. 2417–2431, May 2014.
- [13] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Improving wireless physical layer security via cooperating relays," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1875–1888, Mar. 2010.
- [14] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Trans. on Inform. Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994.
- [15] J. H. Conway and N. J. A. Sloane, *Sphere packing, lattices and groups*. New York: Springer, 1998.
- [16] G. D. Forney Jr., M. D. Trott, and S.-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Trans. on Inform. Theory*, vol. 46, no. 3, pp. 820–850, May 2000.
- [17] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Trans. on Inform. Theory*, vol. 44, no. 1, pp. 273–278, Jan. 1998.
- [18] U. Erez and R. Zamir, "Achieving $\frac{1}{2} \log(1 + \text{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. on Inform. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

- [19] M.-R. Sadeghi, A. H. Banihashemi, and D. Panario, "Low-density parity-check lattices: Construction and decoding analysis," *IEEE Trans. on Inform. Theory*, vol. 52, no. 10, pp. 4481–4495, Oct. 2006.
- [20] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. on Inform. Theory*, vol. 54, no. 4, pp. 1561–1585, Apr. 2008.
- [21] A. Sakzad, M.-R. Sadeghi, and D. Panario, "Construction of turbo lattices," in *48th Annual Allerton Conference on Commun., Control, and Computing (Allerton), 2010*, Sep. 2010, pp. 14–21.
- [22] N. di Pietro, J. J. Boutros, G. Zemor, and L. Brunel, "Integer low-density lattices based on construction A," in *IEEE Inform. Theory Workshop (ITW), 2012*, Sep. 2012, pp. 422–426.
- [23] M.-R. Sadeghi and A. Sakzad, "On the performance of 1-level LDPC lattices," in *Iran Workshop on Commun. and Inform. Theory (IWCIT), 2013*, May 2013, pp. 1–5.
- [24] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. on Inform. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
- [25] N. Yiwei and N. Devroye, "Lattice codes for the gaussian relay channel: Decode-and-forward and compress-and-forward," *IEEE Trans. on Inform. Theory*, vol. 59, no. 8, pp. 4927–4948, Aug. 2013.
- [26] N. S. Ferdinand, M. Norkleby, and B. Aazhang, "Low density lattice codes for the relay channel," in *IEEE International Conference on Commun. (ICC), 2013*, Jun. 2013, pp. 3035–3040.
- [27] M. Norkleby and B. Aazhang, "Lattice coding over the relay channel," in *IEEE International Conference on Commun. (ICC), 2011*, Jun. 2011, pp. 1–5.
- [28] N. S. Ferdinand, M. Norkleby, and B. Aazhang, "Low-density lattice codes for full-duplex relay channels," *IEEE Trans. on Wireless Commun.*, vol. 14, no. 4, pp. 2309–2321, Apr. 2015.
- [29] F. Chen, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Trans. on Inform. Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.
- [30] G. D. Forney Jr. and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *IEEE Trans. on Inform. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.
- [31] Y. Yona and M. Feder, "Efficient parametric decoder of low density lattice codes," in *IEEE International Symposium on Inform. Theory (ISIT), 2009*, Jun. 2009, pp. 744–748.
- [32] H.-A. Loeliger, "Averaging bounds for lattices and linear codes," *IEEE Trans. on Inform. Theory*, vol. 43, no. 6, pp. 1767–1773, Nov. 1997.
- [33] H. Khodaiemehr, M.-R. Sadeghi, and A. Sakzad, "Practical encoder and decoder for power constrained QC-LDPC lattices." [Online]. Available: <http://arxiv.org/abs/1603.07010>
- [34] J. Conway and N. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. on Inform. Theory*, vol. 32, no. 1, pp. 41–50, Jan. 1986.
- [35] S. Johnson, *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*.
- [36] N. Sommer, M. Feder, and O. Shalvi, "Shaping methods for low-density lattice codes," in *IEEE Inform. Theory Workshop (ITW), 2009*, Oct. 2009, pp. 238–242.
- [37] T. M. Aulin, "Breadth-first maximum likelihood sequence detection: basics," *IEEE Trans. on Commun.*, vol. 47, no. 2, pp. 208–216, Feb. 1999.